

1 **CLAIMS**

2

3 1. A method to be implemented in a computer system comprising a

4 processor and a memory, the method for managing a run queue comprising a first

5 plurality of threads sorted with respect to one another based on thread priority, the

6 method comprising:

7 in a deterministic amount of time, associating a second plurality of threads

8 that is priority sorted with the run queue in a manner that maintains a priority

9 based scheduling semantic of the run queue.

10

11 2. A method as recited in claim 1, wherein the second plurality of

12 threads comprises a root thread, and wherein associating the second plurality of

13 threads with the run queue further comprises:

14 inserting only the root thread into the run queue.

15

16 3. A method as recited in claim 1, wherein the associating the second

17 plurality of threads with the run queue further comprises:

18 inserting each thread in the second plurality of threads into the run queue

19 independent of any additional other queue access.

20

21 4. A method as recited in claim 1, wherein associating the second

22 plurality of threads with the run queue further comprises:

23 inserting only a root thread of the second plurality into the run queue to

24 represent the second plurality of nodes.

25

1 5. A method as recited in claim 1, wherein associating the second
2 plurality of threads with the run queue further comprises:

3 inserting only a root thread of the second plurality into the run queue; and

4 wherein the method further comprises:

5 removing the root thread from the run queue; and

6 responsive to removing the root thread, inserting a next thread of the
7 second plurality into the run queue such that the priority based scheduling
8 semantic of the run queue is preserved.

9
10
11 6. A method as recited in claim 1, wherein the method further
12 comprises:

13 inserting a root thread of the second plurality into the run queue;

14 removing the root thread from the run queue for execution; and

15 responsive to removing the root thread and independent of any additional
16 other queue access, inserting a next thread of the second plurality of threads into
17 the run queue.

18
19 7. One or more computer-readable media comprising computer-
20 executable instructions to perform a method as recited in claim 1.

1 **8.** A system for managing a run queue, the run queue comprising a first
2 plurality of threads, each thread in the first plurality having a respective priority,
3 the first plurality being sorted such that a thread having a high priority is removed
4 from the run queue before a thread having a lower priority, the system comprising:

5 a memory for storing the run queue and computer-executable instructions;

6 a processor operatively coupled to the memory, the processor being
7 configured to execute the computer-executable instructions for:

8 in a deterministic amount of time, associating the second plurality of
9 threads that is priority sorted with the run queue, the associating maintaining a
10 priority based scheduling semantic of the run queue.

11
12 **9.** A system as recited in claim 8, wherein associating the second
13 plurality with the run queue is performed independent of more than a single other
14 queue access.

15
16 **10.** A system as recited in claim 8, wherein the second plurality
17 comprises a root thread operatively coupled to one or more other threads of the
18 second plurality, each of the one or more other threads having a respective priority
19 that is a lower priority or an equal priority as compared to a priority of the root
20 node.

21
22 **11.** A system as recited in claim 8, wherein associating the second
23 plurality of threads with the run queue further comprises:

24 inserting only a root thread of the second plurality into the run queue.
25

1 **12.** A system as recited in claim 8, wherein associating the second
2 plurality of threads with the run queue further comprises:

3 inserting only a root thread of the second plurality into the run queue to
4 represent the second plurality of threads.

5
6 **13.** A system as recited in claim 8:

7 wherein the first plurality of threads is a first linked list data structure;

8 wherein the second plurality of threads is a second linked list data structure
9 comprising a root node that is operatively coupled to one or more other threads in
10 the second plurality; and

11 wherein the single insert operation is an operation comprising inserting the
12 root node into a position in the first linked list data structure.

13
14 **14.** A system as recited in claim 8, wherein associating the second
15 plurality of threads with the run queue further comprises:

16 inserting only a root thread of the second plurality into the run queue; and

17 wherein the method further comprises:

18 removing the root thread from the run queue; and

19 responsive to removing the root thread, inserting a next thread of the
20 second plurality into the run queue such that a priority based scheduling semantic
21 of the run queue is preserved.

22
23 **15.** A system as recited in claim 8, wherein the processor is further
24 configured to execute computer program instructions for:

25 inserting a root thread of the second plurality into the run queue;

1 removing the root thread from the run queue for execution; and
2 responsive to removing the root thread and independent of any additional
3 other queue access, inserting a next thread of the second plurality into the run
4 queue.

5
6 **16.** A computer-readable storage medium comprising computer-
7 executable instructions to manage a run queue sorted with to one another based on
8 thread priority, the computer-executable instructions comprising instructions for:

9 in a deterministic amount of time that is independent of the number
10 of threads in a second plurality of threads that is priority sorted, associating the
11 second plurality of threads with the run queue in a manner that maintains a priority
12 based scheduling semantic of the run queue.

13
14 **17.** A computer-readable storage medium as recited in claim 16,
15 wherein the second plurality of threads comprises a root thread that is operatively
16 coupled to one or more other threads of the second plurality, and wherein the
17 instructions for associating further comprise:

18 inserting only the root thread into the first plurality of threads to represent
19 the second plurality of threads.
20
21
22
23
24
25

1 **18.** A computer-readable storage medium as recited in claim 16,
2 wherein the first plurality of threads is a first linked list data structure, the second
3 plurality of threads is a second linked list data structure comprising a root node
4 that is operatively coupled to one or more other threads in the second plurality,
5 and the deterministic amount of time is a result of a single insert operation to
6 insert the root node into the first linked list data structure.

7
8 **19.** A computer-readable storage medium as recited in claim 16,
9 wherein the instructions for associating further comprise:

10 inserting only a root thread of the second plurality of threads into the first
11 plurality of threads;

12 and wherein the computer-executable instructions further comprise
13 instructions for:

14 removing the root thread from the run queue; and
15 responsive to removing the root thread, inserting a next thread of the
16 second plurality into the first plurality in a manner that maintains a priority based
17 scheduling semantic of the run queue .

18
19 **20.** A computer-readable storage medium as recited in claim 19,
20 wherein the acts for inserting the next thread are performed independent of an
21 other queue.

22
23 **21.** A computer-readable storage medium as recited in claim 16,
24 wherein the instructions for associating further comprise instructions for:

25 inserting a root thread of the second plurality into the first plurality;

1 removing the root thread from the first plurality for execution; and
2 responsive to removing the root thread, inserting a next thread of the
3 second plurality into the first plurality independent of any additional access to
4 another different queue.

5
6 **22.** A computer comprising a processor operatively coupled to a
7 computer-readable medium as recited in claim 16, the processor configured to
8 execute the computer-executable instructions.

9
10 **23.** A run queue data structure comprising:
11 a first dimension data field comprising a first plurality of threads sorted
12 with respect to thread priority; and
13 a second dimension data field comprising a second plurality of threads
14 sorted based on thread priority, the second plurality of threads comprising a root
15 thread and one or more other threads.

16
17 **24.** A computer-readable medium comprising a run queue data structure
18 as recited in claim 23.
19
20
21
22
23
24
25